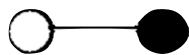


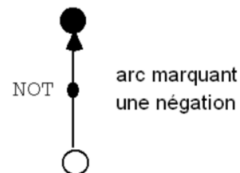
Représentation NETL

- Noeuds et arcs

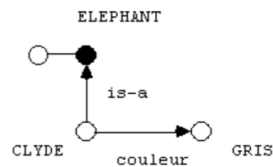
- Noeud représentant un **concept** et l'ensemble des **individuels** correspondant à ce concept



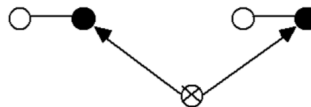
- Négation



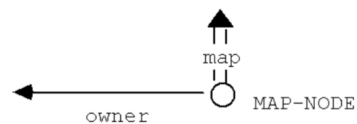
- Propagation



- Détection d'incohérences



- Notion de rôle



Représentation LISP

noeud ((name . #) (type . #) (inargs . #) (outargs . #))

arcs ((type . #) (from . #) (to . #))

EX: N0 ((name . Roxane) (type . individu) (inargs . A0) (outargs . A1))

N1 ((name . Christiane) (type . individu) (inargs . A1) (outargs . A0))

A0 ((type . aime) (from . N0) (to . N1))

A1 ((type . aime) (from . N1) (to . N0))

(gentemp "N") ==> N0 N1 N2 N3

```

(defvar *noeuds* nil)
(defvar *arcs* nil)
(defun defnode (name type)
  (let ((N (gentemp "N")))
    (set N (list (cons 'name name) (cons 'type type)) ) ;;
    ici utilise set on met juste N dans *noeuds*, si utilise setq
    il va push la liste dans *noeuds*.
    (pushnew N *noeuds*)
    N
  )
)

```

- Mon solution pour définir ARCS, j'ai pas réalisé la partie du mise à jour.

```

(defun defarc (type from to)
  ;; si les noeuds from et to n'existent pas -> erreur
  ;; creer de l'arc
  ;;mise a jour de in-arcs et out-arcs dans from et to
  ;;maj de *arcs*
  ;;retourne l'id de l'arc
  (if (null from) (error "Noeud vide...")) ; gestion des
erreurs
  (if (null to) (error "Noeud vide...")) ; gestion des
erreurs
  (let ((A (gentemp "A")))
    (set A (list (cons 'type type) (cons 'from from)
      (cons 'to to) ) )
    (pushnew A *arcs*)
    A
  ) )
(defnode 'LIU 'Man)
(defnode 'Yan 'Man)
(defarc 'aime N1 N2)
(print A3) Ici j'ai une question, dans ce cas gentemp conter
continuer, il est pas recommencé à 1???
(print N1)
(print N2)

```

Version du cours:

```
(defun defarc (type from to)
  (cond ((not (and (member from *noeuds*) (member to
*noeuds*)))
    (error "Noeud non existant")
  )
  (T (let ( (A (gentemp "A")))
    (set A (list (cons 'type type) (cons 'from
from) (cons 'to to) ))
    (push A *arcs*)
    (ajouter_arc_noeud A from to)
    A
  ))
)
)
```